PCT
WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06K 9/32, 9/68

A2

(11) International Publication Number: WO 90/03012

(43) International Publication Date: 22 March 1990 (22.03.90)

(21) International Application Number: PCT/GB89/01043

(22) International Filing Date: 6 September 1989 (06.09.89)

(30) Priority data:
8821024.0 7 September 1988 (07.09.88) GB

(71)(72) Applicants and Inventors: ETHERINGTON, Harry, James [GB/GB]; "Tallarna", Grosvenor Road, Godalming, Surrey GU7 1PA (GB). JOSLIN, Paul, Carter [GB/GB]; 39 The Oaks, Bracknell, Berks RG12 2XG (GB). NEWMAN, Roger, Keith [GB/GB]; Meadow Croft, Hoe Road, Bishops Waltham, Southampton, Hampshire SO3 1DU (GB).

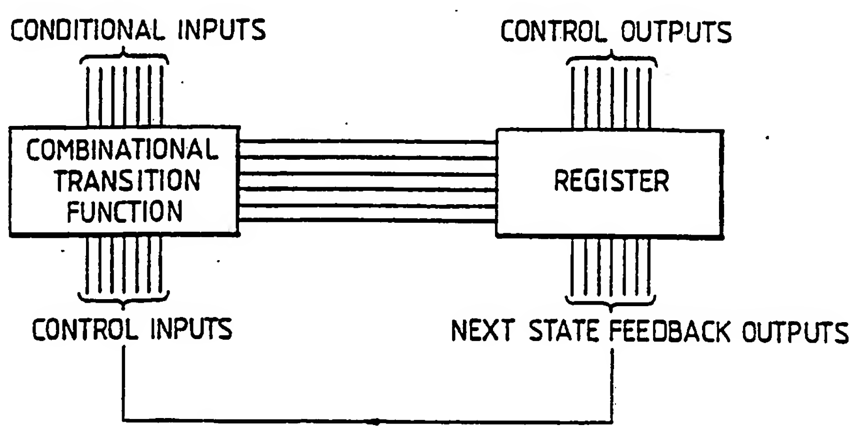
(72) Inventors; and
(75) Inventors/Applicants *(for US only)* : BAXTER, Peter [GB/GB]; 85 Lowdon Close, Badger Farm, Winchester, Hants SO22 4EW (GB). LIDSTONE, Steven [GB/GB]; 37 Nutbeem Road, East Leigh, Hants SO5 5JP (GB).

(74) Agent: GILL JENNINGS & EVERY; 53/64 Chancery Lane, London WC2A 1HN (GB).

(81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), FR (European patent), GB (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent), US.

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: IMAGE RECOGNITION



(57) Abstract

The invention relates to an optical character recognition system in which video data defining an image is fed to a bit map (7) for subsequent processing. The image in the bit map is segmented and classified using an N-tuple processor. The segmentation and classification are synchronous-state machine implemented.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MR	Mauritania
BE	Belgium	GA	Gabon	MW	Malawi
BF	Burkina Fasso	GB	United Kingdom	NL	Netherlands
BG	Bulgaria	HU	Hungary	NO	Norway
BJ	Benin	IT	Italy	RO	Romania
BR	Brazil	JP	Japan	SD	Sudan
CA	Canada	KP	Democratic People's Republic of Korea	SE	Sweden
CF	Central African Republic			SN	Senegal
CG	Congo	KR	Republic of Korea	SU	Soviet Union
CH	Switzerland	LI	Liechtenstein	TD	Chad
CM	Cameroon	LK	Sri Lanka	TG	Togo
DE	Germany, Federal Republic of	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		

IMAGE RECOGNITION

The invention relates to methods and apparatus for recognising two dimensional images, such as text characters, represented in binary form as a bit map of pixels.

Various character recognition systems have been developed and proposed and these systems generally fall into two types:

1. Template (mask) matching or Matrix matching: where the image of the character is compared with a set of stored prototype images to achieve a match and recognise the character. The technique is constrained by the amount of computer memory required to store the different fonts, it requires the character font to be known to the product, it requires well-defined characters, it does not learn from its mistakes.

Where a good match cannot be expected, the product costs increase with:

a. pre-processing to remove distortions.
b. post-processing to assess the degrees of match to the prototype templates

2. Topological (Topographical) analysis or Shape (feature) analysis: where the shape and features of a character image are examined in order that an algorithmic match may be attempted. Such a technique has a high degree of font independence and it has a learning capability. Problems exist with poorly defined, distorted or broken characters (images) such as are met with in every-day print since these distortions affect the features by which the character is to be recognised.

Software means are predominately used to perform topological analysis. Thus the recognition speeds tend

to be low, in order to restrain the product costs; since the recognition speed is dependent on the execution times of the recognition computer system the faster the recognition speed, the more powerful the computer system (that is required), the greater the product costs.

Techniques have been developed based on so-called N-tuple classifiers which were originally described in a paper entitled "Pattern Recognition and Reading by Machine" by Bledsoe and Browning, 1959 Proceedings of Eastern Joint Computer Conference, pages 225-232 and which are also described in "Guide to pattern recognition using random-access memories" by Aleksander and Stonham, 1979. Computers and Digital Techniques Vol. 2, No. 1, pages 29 -40. The N-tuple method is essentially a means comparing information presented to a system with information already "learnt" by the system, so that the system can then make "most like" decisions. This methodology has an ability to cope with the recognition of patterns and shapes including multi-font character recognition. It does not require the font (to be recognised) to be pre-determined, it does however require an adequate training, over a sufficient range of fonts and over a sufficient range of distortions within a font to be able to discriminate between characters, for those fonts which it is likely to be required to recognise. Examples of patent specifications illustrating these N-tuple techniques are GB-A-1296701, GB-A-1431438, and GB-A-2112194. These systems achieve improved recognition results over the previous types of pattern recognition systems but require either, very expensive but fast hardware based systems or, lower priced (but still expensive), slow software based systems.

In accordance with one aspect of the present invention, image recognition apparatus comprises a first synchronous state machine for segmenting a number of

images defined in bit map form into separate pixel groups; and a second synchronous state machine to which each pixel group is applied for classification.

The intention is that each pixel group found by the segmenting state machine will correspond to an image which can be classified.

The inventors have realised the unique merits of the N-tuple method in coping with the variable print quality of "real-world" documents.

The inventors have further realised that the problems associated with the previous application, of the N-tuple method, could be overcome by the design approach. These problems are the inter-relationships of the speed of recognition and the product costs, that is slow operation, high expense.

An important feature of the invention is that the inventors have realised the unique and significant advantages in using a synchronous state machine approach, to implement much of the core technology recognition function. This approach is particularly advantageous when utilising a technology development based on the N-tuple method of pattern recognition.

The core technology recognition function comprises:

(a) Segmentation. The process of breaking the scanned information into separate distinct images i.e. the process of shape extraction. The segmentation process is coupled with: Registration. The process of providing positional information, to register the relationship of the individual segmented images, thereby allowing the "recognised" characters to be assembled into a data-stream to an appropriate format.

(b) Classification. The process of classifying the images into pre-defined classes.

The classification process includes the means of handling cases where the classifier:

(i) is unable to make a true decision, i.e. a reject error; in this case the classifier should label the result accordingly.

(ii) makes a wrong decision, i.e. a substitution error; in this case the system has to recognise the error from other information such as the context.

Segmentation and classification are described in IBM Journal of Research and Development, vol. 27, No. 4, p.p. 386-399.

A synchronous state machine is one where the stages, of the processes, are stepped-on simultaneously, under control of a system clock. Thus the time penalties may be avoided which can occur in asynchronous machines, associated with the processing of each stage, for example due to interrupt routines, polling routines hand-shaking routines and the like.

A state machine approach, for this image recognition application of the N-tuple method of pattern recognition, allows the use of a hardware implementation. This allows a much higher speed of image recognition (than can be achieved by predominately a software approach) at the moderate product prices which are associated with the software based products.

In accordance with a second aspect of the present invention, a method of recognising images represented by respective digital pixel groups comprises presenting each pixel group to an N-tuple classifier having a number of discriminators each adapted to recognise a respective class of a predetermined group of classes and is characterised in that each pixel group is presented to the discriminators in a predetermined sequence; and in that as soon as the output from a discriminator satisfies

a recognition condition, the presentation of the pixel group to the classifier is terminated.

In accordance with a third aspect of the present invention, apparatus for recognising images represented by respective digital pixel groups comprises an N-tuple classifier including a number of discriminators each adapted to recognise a respective class of a predetermined group of classes and to which the pixel groups are presented, the apparatus being arranged to present each pixel group to the discriminators in a predetermined sequence; and recognition means for monitoring the output of the discriminators and for terminating the presentation of the pixel group to the classifier as soon as the output from a discriminator satisfies a recognition condition.

For the first time, we have realised that it is possible to make operation of an N-tuple classifier interactive with the recognition process so that as soon as a character is sufficiently identified, further operation of the classifier is terminated.

In one example, the method comprises comparing the output from each discriminator with a threshold, the recognition condition being satisfied when the threshold is exceeded. Typically, the situation is:

(i) There will be some threshold 'A' above which, the image is identified (recognised).

(ii) There will be some threshold 'B' below which, the image is not immediately recognised.

(iii) The band between 'A' and 'B' for which the image is recognised as belonging to a group of classes, for example lower case o e c, but further processing is required to allow the particular image to be recognised.

(iv) In the event that the discriminator response

"scores" are less than B this would mean that the classifier has performed a complete operation, in that case the Ranking Order of scores is examined and should the difference between the maximum discriminator output and the next discriminator outputs satisfy predetermined criteria, then the recognition condition is satisfied, i.e. the character is recognised (as the highest score).

In another example, each pixel group may be presented to the discriminators in the order of frequency of occurrence of the classes represented by the discriminators. For example, where the images comprise text characters originating from an English language text, the pixel groups may initially be presented to discriminators representing the vowels (as being the commonest occurring letters in the English language) and subsequently to other groups of classes having successively decreasing frequencies of occurrence.

In a further arrangement, the discriminator or discriminators to which each pixel group is applied may be chosen in accordance with the location of the pixel group defining the images within the context of the previously detected images. For example, in the case of text, if a full stop has been detected then it would be expected that the next letter is upper case and thus the next pixel group will be presented initially to the group of classes defining the upper case letters.

The concept of interaction between the classifier and the recognition process is also utilised in a method according to a fourth aspect of the invention for recognising images represented by respective digital pixel groups, the method comprising presenting each pixel group to an N-tuple classifer having a number of discriminators each adapted to recognise a respective class of a predetermined group of classes characterised

in that if none of the discriminator outputs satisfies a recognition condition but it is determined that the pixel group defines an image falling within a group of the classes, the method further comprises presenting a portion of the pixel group to a subsidiary N-tuple classifier having a number of subsidiary discriminators each adapted to recognise a respective portion of the group of classes.

In the case of the English language, certain letters such as "o", "e", and "c" have similar forms and the classifier may not have been trained sufficiently to distinguish between them. However, if the right-hand half of each of those letters is compared, these are significantly different and thus by training a subsidiary classifier on the right-hand halves alone, these particular characters can be distinguished relative to each other.

In accordance with a fifth aspect of the invention, apparatus for recognising images represented by respective digitial pixel groups comprises an N-tuple classifier having a number of discriminators each adapted to recognise a respective class of a predetermined group of classes and to which each pixel group is presented; recognition means for monitoring the outputs of the discriminators; and a subsidiary N-tuple classifier having a number of subsidiary discriminators each adapted to recognise a respective class of a predetermined group of classes defining portions of a respective group of images, the recognition means being adapted to present a portion of a pixel group to the subsidiary classifier if it is determined that the discriminator outputs do not satisfy a recognition condition but the discriminator outputs define an image falling within the group of classes.

In all these cases, the method preferably further comprises storing data defining the recognised class of the image represented by the pixel group for which purpose the apparatus preferably further comprises storage means.

Typically, in order to reduce processing time, each pixel group is presented simultaneously to groups of two or more discriminators in the classifier and, where appropriate, the subsidiary classifier.

In this specification the bit map described will generally have a data bus width of one bit. The (image) processing tasks require access to a memory system in an incremental fashion and because this is a pixel by pixel addressable task, using dedicated logic circuits, it is more efficiently organised with the memory (bit map) organised with a data bus width of one bit.

In this connection, in order to utilise commercially available, low cost, memory devices and also to use commercially available, low cost, microprocessor devices, the inventors have recognised that the memory system could (with advantage) be organised as a dual port system:

(1) The first port being a conventional memory access port designed to suit a particular microprocessor bus, e.g. an eight bit wide data bus.

(2) The second port being organised to have a data bus width of one bit, with an addressing system providing for an incremental addressing system allowing for both positive and negative displacements in two axes, since it is desired to access individual pixels stored in a two dimensional array.

It is important in both conventional N-tuple classification and the improvements to that classification described above, to be able to present to

the classifier accurately segmented pixel groups which are known to define a single image, such as a character. In the case of printed text, segmentation is complicated by the fact that individual characters are not always spaced evenly from adjacent characters. For example, proportionally spaced characters have a variable spacing and certain characters such as the letter pair "fo" may overlap. These and related problems are outlined in the IBM reference mentioned above.

In accordance with a sixth aspect of the present invention, a method of segmenting images represented in bit map form comprises scanning the bit map to determine the maximum extents of an image in first and second orthogonal directions and recording for each scan line in the first direction the coordinates of the extreme pixels of the image in the second orthogonal direction; and selecting as defining an image only those pixels within a rectangle defined by the previously determined extents and falling within the previously determined extreme pixel coordinates.

In accordance with a seventh aspect of the present invention, apparatus for segmenting images represented in bit map form comprises scanning means for scanning the bit map to determine the maximum extents of an image in first and second orthogonal directions and for recording for each scan line in the first direction the coordinates of the extreme pixels of the image in the second orthogonal direction; and selection means for selecting as defining an image only those pixels within a rectangle defined by the previously determined extents and falling within the previously determined extreme pixel coordinates.

This method and apparatus is able to cope with overlapping and proportionally spaced images. In the case of touching characters the pixel group comprising

the image block, is divided into sub-blocks by an estimation of the character boundaries within the pixel block, (representing two or more character pixel groups), for example by a knowledge of the character aspect ratio (obtained from an histogram analysis of the text), each sub-block is then submitted separately to a classification process.

Typically, in the case of a page of text, the scanning of the bit map is carried out in a series of horizontally spaced, vertical scan lines and this leads to the ability to compensate for skew from a knowledge of the line spacing or pitch deduced from a histogram analysis of the page of text.

Preferably, the selecting step comprises scanning the bit map in a series of lines extending in a second orthogonal direction and spaced apart in the first orthogonal direction, each line having a length corresponding to the distance between the respective extreme pixel coordinates.

Some previous segmentation methods have involved locating a black pixel and then examining the immediate neighbours to that pixel and subsequently locating one of the adjacent pixels which is black and repeating the process. This leads to considerable duplication in that the same pixels will be examined several times and thus segmentation is a relatively slow process.

In accordance with an eighth aspect of the present invention, a method of segmenting images represented in bit map form comprises

a) scanning the bit map to detect a shape which may comprise an image;

b) recording the location of those pixels in the bit map which define the detected shape;

and repeating the steps a) and b) to locate other images while ignoring in step a) each pixel whose location has been recorded in a step b).

In accordance with a ninth aspect of the present invention, apparatus for segmenting images represented in bit map form comprises scanning means for scanning the bit map to detect a shape which may comprise an image; and a memory for recording the location of those pixels in the bit map which define the detected shape, whereby the scanning means only responds to those pixels of the bit map whose locations have not been recorded in the memory.

Typically, step b) comprises providing a second bit map coterminous with the bit map defining the images, and recording in the second bit map those pixels which have been found during the scanning step to correspond to a detected shape.

Preferably, means are provided to ignore isolated black pixels, during the scanning process, as unwanted background noise. An isolated black pixel is one where all its (eight) neighbours are white pixels.

The images with which the invention is concerned may include characters, such as text characters (numbers and alpha characters) both arabic and non-arabic, and also other two dimensional predetermined patterns and shapes as for example obtained by robot manipulators carrying video cameras.

The bit map defining the images may be generated in any conventional manner such as by means of a CCD array, a video scan and subsequent digital processing and the like.

Particularly advantageous methods and apparatus are constituted by combinations of the first to ninth aspects of the invention.

An example of a character recognition system according to the invention will now be described with reference to the accompanying drawings, in which:-

Figure 1 illustrates the overall system;

Figure 2 illustrates the construction of the recognition system;

Figure 3 is a flowchart illustrating the operation of the computer control system;

Figure 4 is a block diagram of the image preprocessing circuit;

Figure 5 illustrates the memory system;

Figure 6 is a block diagram of the scan-search circuit

Figure 7 illustrates the segmentation system;

Figure 8A-8D illustrates the extraction process;

Figure 9 illustrates the datum conditions for an extracted shape;

Figure 10A-10B illustrates the scaling and normalising functions;

Figure 11 shows an example (in block diagram form)of a variable scaling system,

Figure 12 illustrates an example of a scaling table;

Figure 13A-13B illustrate an an example of N-tuple mapping;

Figure 14 illustrates the classification system;

Figure 15 is a flow chart illustrating the operation of the classification system;

Figure 16 illustrate a combinational transition function.

Figure 1 illustrates the OCR (optical character recognition) system, by which indicia, existing in printed or written form, are captured as images and converted to data encoded to a computer industry standard.

A video scanner (1) produces digitised video data, representing the black or white pixel image data, captured from a scanned page, as a line by line sequence of the indicia on that page. A scanner video interface (2) orders the video data into a form for subsequent data processing. The video data is sent to a recognition unit (3), the recognition unit output (4) being the indicia (character) data encoded to a suitable computer industry standard, such as ASCII (American Standard Code for Information Interchange).

The scanner (1) may be any convenient form of commercial optical scanner, with appropriate product capabilities and facilities, that is a paper-handling capability (sheet-feed or flat-bed) at an appropriate image resolution, at an appropriate scan time for a page. Commercial scanners generally have an image resolution of 300 dots per inch (dpi), which is adequate for most OCR purposes. Commercial scanners are available with scan times of less than 3 seconds for an ISO standard A4 page size, which allows for a high speed of character recognition circa 1000 characters per second. The scanner video interface (2) may take one of several forms, serial or parallel, e.g. SCSI (Small Computer Systems Interface).

The scanner (1) may alternatively be constructed (as a page scanner) by utilising for example a full A4 width CCD (charge coupled device) photo element array,

coupled with A control system consisting of analogue image data capture, thresholding, digital conversion, timing, scan control and interface circuits; this to provide digitised video data, representing the bit image data captured from the scanned page.

Figure 2 illustrates the overall construction of the recognition unit (3). The recognition unit system provides for the segmentation and classification functions, these functions being the processes of breaking the scanned images into separate distinct images for each character, registering the relationship of the individual (segmented) character images and classifying the character images into pre-defined character classes.

The video data from the scanner (1) is interfaced into the recognition unit (3) by a video interface (5), this video interface (5) may be any of several known forms, to suit the scanner's video interface (2). The video data is fed into an image pre-processing circuit (6), which processes the video data into a RAM (random access memory), in the form of an image bit map (7) having a one bit wide data bus.

The image bit map (7) operates in conjunction with a shadow bit map (8), having pixel locations in one to one correspondence with the image bit map (7). The shadow bit map (8) is used to avoid processing the same pixels several times, such duplicated processing occurs in some known segmentation schemes.

A scan-search circuit (9) performs a vertical raster scan of the image bit map (7),starting from the top left hand corner of the "page". This is to locate potential characters by searching for black pixels which have not been previously processed, i.e. to

search for black pixels not in the shadow bit map (8). A synchronous state machine segmentation system (10) is used to extract the character shape, associated with the found black pixel. The extracted character shape is fed into a normalise and randomise system function (11). The character shape is, by this function (11), normalised in size and converted into a random N-tuple form, which is then loaded into the buffered input of a synchronous state machine classification system (12). The classification system (12) identifies (classifies) each character, that is so presented. The identification of the character is fed into the computer control system (13) for post-processing. The computer control system (13) is also used to control certain aspects of the operation of the recognition unit (3). The computer control system (13) uses a commercial micro processor which is software controlled, the mode of operation is illustrated in Figure 3.

Output of the character data to the Host system is via the system interface (14).

The construction of the recognition unit (3) shown in Figure 2 will now be described in more detail with reference to Figures 3 - 16.

The video data received from the scanner (1), is fed into the image preprocessing circuit (6), via the video interface (5), as initiated by the computer control system (13) (Steps 101 and 102 Fig.3).

The image pre-processing circuit (6) is shown in more detail in Figure 4. The video data is fed to control logic (15). Dependent on the OCR application and dependent on the resolution of the scanner (1), it may be desired to compress the video data, from say 400 dpi to say 200 dpi. If data compression is necessary,

the control logic (15) will feed the video data to the compression circuits:- horizontal compression (16) and vertical compression (17); for the example quoted both of these compression circuits would be 2:1. The data compression may be arranged to favour white, to enhance the character bit image separation. A circuit (18) electronically adds a white border (to define the boundary conditions to be used for subsequent scanning of the bit map) to the compressed (or otherwise) video data, at the time that this video data is being written into the image bit map (7); at the same time the shadow bit map (8) having a one bit wide data bus is cleared to white (step 103 Fig.3). The process continues until the video data is completely received or until the image bit map is full, any unfilled portion of the image bit map being written to white. If the scanned video data exceeds the capacity of the image bit map (7), then the video data will require to be loaded [into the recognition unit (3)] in more than one data transfer operation, this will be controlled by the computer control system (13). At the completion of the setting-up the bit maps (Step 104 Fig.3) the computer control system (13) sets bitmap pointers to the scan "start" position, (Step 105 Fig.3).

Commercially available memory devices, which could be used to constitute the image and shadow bit maps (7) and (8), have all been developed for convenient use with commercial microprocessors. Such memory devices have memories organised with a data bus width which suits a particular micro-processor standard, the commonly encountered data widths being eight, sixteen or thirty-two bits. In the present application the memory system is associated with the processing of shapes, (image data) contained within that memory;

the data to be processed exists as black or white pixels (binary values of picture elements) which are stored in a two dimensional array of single pixel values and which is referred to as a bit map. The image data processing tasks require access to the bit map memory system in an incremental fashion, which is a pixel by pixel addressable task. This task is most efficiently organised with the bit map having a data width of one bit, since dedicated mixtures of combinational and sequential logic can then be used to achieve higher execution speeds, than could be achieved by a conventional microprocessor access to memory via a multi-bit data bus and software selection.

Figure 5 illustrates in more detail the organisation of the memory system, which is designed to use commercially available (low cost) microprocessor and memory devices and which is constructed as a dual port system. The first port representing the microprocessor interface (19) is a conventional memory access port, designed to suit a particular microprocessor data bus width, for example using an eight bit wide data bus. The second port representing the image processing interface (20) is organised to have a one bit wide data bus, the addressing system of which provides for an incremental addressing, allowing positive and negative movements in the two axes of the memory plane. Such movements (in the two axes of the memory plane) are required in the segmentation process to be described.

An access arbitration circuit (21) prevents a memory access occurring simultaneously from both the microprocessor and the image processing interfaces; the microprocessor interface (19) is one that can be forced to wait until the memory is ready for it. The access arbitration logic ensures that only one set of address

and data drivers are energised at any one time, thus preventing an undesirable "clash" of accesses. A signal M/S is used to enable one or other set of drivers, within the address multiplex and write multiplex circuits(22) and (23). The address multiplex circuit (22) functions as a selecting switch, according to which interface had the right of access to the memory at any particular time, it allows for the selection of the appropriate address required for any particular memory access. The write multiplex circuit (23) functions in a similar way, as the address multiplex circuit (22).

The data organisation of the memory, is such that it it presented to the microprocessor in the familiar eight bits (byte) format, that is used by most micro computer memory systems. Each byte of the memory array (24) comprising four bits of the image bit map (7) and four bits of the shadow bit map (8). A 1 of 8 write decoder (25) is required to enable the image processing interface (20), which handles one pixel at a time, to selectively write one bit within the eight bits of a byte. An equivalent function is required for read access be the memory from the image processing interface (20), in order for this interface to be able to select one particular bit from the byte, this is referred to as the 1 of 8 bit select circuit (26).

The 8 bit datadriver (27) does not require the complexity that would be normally required if a bit selecting interface was to be provided to allow the image processing functions to write bits to memory on an individual basis. This simplification is achieved because,

(a) the use of one bit wide memory devices allows the image processing functions to "read from" and "write to" memory on a single bit basis, i.e. there is no need

for a "read and rewrite" operation that would be necessary to perform a single bit write operation if 8 bit wide memory devices were used,

(b) the image processing functions require a write only function of logic "1's" (representing black pixels) to the shadow memory.

The address register (28), in conjunction with the offset address adder (29), controls the addressing of the memory array (24). The address register (28) holds coordinate pixel location information, corresponding to the coordinates of the pixels representing the scanned image video data. The offset address adder (29) is a binary parallel adder circuit,constructed from combinational logic; it is capable of handling an X or Y offset in positive and negative form in order to be able to address any pixel within the memory array (24). The addressed pixel can be either (a) left or right of the horizontal coordinate stored in the address register (29) or (b) above or below the vertical coordinate stored in the address register(29). Negative values (for left and above) are handled by treating the X and Y addresses as two's complement binary numbers. The X and Y addresses presented from the image processing interface (20) need only to address a limited (256 x 256 pixels) area of the memory array (29), this is because these addresses are used for the character segmentation, which requires only sufficient memory space for each character shape one at a time.

The address register (28) performs various functions, independent on the operational aspects of the memory system: (a) In the setting-up of the image bit-map (7) the address register (28) counts through the XY coordinate addresses of the bit map to allow the storage of the pixel data, black or white,

corresponding to that coordinate address; for this operation the XY offset values are set to zero in the offset address adder (29). b) In the segmentation process the "found coordinate" of the character shape to be segmented, is entered into the address register (28) and the positive or negative movements in X and Y, necessary for the segmentation, are controlled by the offset adder (29). Coincident with the segmentation process the shadow bit map (8) is "read from" and "written to", the shadow bit map (8) being initially cleared to zero (all white) which is taken to be the not yet processed state of the scanned image video data. Writing to the shadow bit map (8) occurs as the image bit map (7) is conditionally scanned during the segmentation process; that is,to segment the character, the data in the image bit map (7) is read and an identical pixel is written to the shadow bit map (8), thus a copy of the character shape will appear in the shadow bit map (8) indicating that the character shape has been segmented. This conditional scanning of the image bit map (7), to ignore character shapes previously encountered and segmented, can be simply achieved by scanning for image bit map pixels which have corresponding zeros (white pixels) in the shadow bit map (i.e. not previously seen pixels), this is implemented with a two input logic gate. An advantage of the shadow bit map (8) is that the image bitmap (7) is preserved to allow a re-examination of the pixel data, if so desired.

A transceiver (bidirectional TRANS-mitter and re-CEIVE 30), present in the data path from the memory (24) to the microprocessor interface (19), isolates the memory from other data circuits connected to the micro processor.

Referring to Figure 3, the next.step 106 is to

initiate the scan-search routine. The image bit map (7) is processed by the scan-search circuit (9), shown in more detail in Figure 6. This process operates in conjunction with the segmentation system (10) (to be described).

The scan process applies a vertical raster scan to the image bit map (7), starting at the top left hand corner (relative to the lines of text on the original scanned document), this position is easily determined relative to the "white border" applied by the image pre-processing circuit (6). The raster scan is in the vertical direction downwards, moving left to right and continues until the scan encounters a non-processed "new" pixel, that is a black pixel which does not appear in the shadow bit map (8). The first "new" (black) pixel of a character so encountered, by the vertical scan, will be the uppermost-leftmost black pixel of that character, the XY position of which will be described as the "found coordinate" of that character.

The vertical raster scan allows for the handling of skewed lines of text (on the original scanned document), since each character is found in sequence according to the spacings of the lines of text. Knowing the range of spacings of the lines of text and the vertical coordinates of each character then the text may be reconstructed on a line by line basis. The vertical spacings may be easily determined from the character positional information derived from the scan process.

As the vertical raster scan of the image bit map (7) proceeds, the shadow bit map (8) is scanned pixel by pixel at the same time. The logic state binary '0' (white) or binary '1' (black) of the pixels in the shadow map, indicating whether or not the pixel in the image map currently being accessed has previously been

accessed, i.e. a binary '0' in the shadow map would indicate a "new" pixel.

The comparison of the binary states of the pixels between the two bit maps is implemented with a 2 input logic gate circuit known as the new pixel selector circuit (31).

As a "new" pixel of a shape is encountered (found) the "found coordinate" of that (found) pixel is loaded into a found coordinate register (32) and a message is sent to the computer control system (13) (step 107, Fig.3). The computer control system (13) immediately starts the segmentation process (to be described) to extract the character shape (step 108, Fig.3). When the character shape is determined (step 109, Fig.3) it is possible to continue the raster scan, since the shadow map is then completed with respect to the "found" character. The scan-search process continues until the end of the image bit map (7) is reached (step 110, Fig.3).

The use of the shadow bit map (8) provides the following benefits.

(a) The image bit map (7) is unaltered. This is a particular benefit where a re-examination of the image data may be required, such a re-examination may be achieved by either (a) re-scanning the image bit map (7) as a whole, or (b) by re-scanning selected areas by clearing the appropriate areas of the shadow bit map (8) back to zero (white) to allow for the pattern(s) to be found again.

(b) Patterns within the image bit map (7) can be of unknown quantity, location and size. The shadow bit map (8) ensures that previously processed groups of pixels corresponding to the patterns already extracted are not reprocessed.

The segmentation process is initiated, at step 108 Fig.3, by the computer control system (13). As previously mentioned the segmentation system (10) uses a synchronous state machine (33) (Figure 7), where combinational transition functions (34) (Fig.7) are used to define the conditions and sequence of the state machine.

A synchronous state machine is one where every stage of the machine is stepped-on on simultaneously under the control of a system clock. Thus avoiding the time penalties which can occur in asynchronous machines associated with the processing of each stage, e.g. interrupt routines, polling routines, hand-shaking routines and the like.

Figure 16 illustrates the use of a combinational transition function (34), which allows for conditional decisions to be made at every step and acts as a combinational logic array, to set the conditions and hence decide the "next state" out of the state register. The function inputs are "conditional" and "feedback"; the outputs are "control" and "next state" feedback. The total number of "states" defines the number of bits in the "next state" feedback path. The combinational transition functions (34) may reside either (a) in a non-volatile memory such as PROM (programmable read only memory), PAL (programmable array logic) etc. or (b) in a volatile memory RAM (random access memory), which is initialised by software on power-up of the machine.

The use of combinational transition functions is particularly advantageous for this application, due to their ease of implementation and modification, as compared to a "logic gate" implementation for the segmentation system.

The segmentation system (10) is shown in more detail in Figure 7. The extraction of the character shape is achieved by the state machine (33) operating in conjunction with the image bit map (7) and shadow bit map (8). The process starts with the "found" pixel of that character shape, the initial condition being with the image bit map (7) XY address pointing to the found coordinate.

The technique used to extract the character shape and determine its boundary conditions will be explained in connection with the letter pair "fo" shown in Figure 8A. In the figure this pair of characters are shown as overlapping, in order to illustrate that the method of segmenting copes with overlapping characters. The character overlap situation can be seen most clearly from Figure 8B, where enclosing rectangles (defined to completely include each character) are illustrated and it will be seen that each rectangle includes a part of the other character. The technique used to define the extent of the character 'f' is to perform an iterative search for the outer edge of the character, i.e. to find the boundary between the black pixels of the character and the white pixels surrounding. Starting at the black pixel corresponding to the found coordinate, the search proceeds around the outside of the boundary and finishes when the start pixel (i.e. the found coordinate) has been returned to. Whilst this search is occurring two measurements are taken, (a) for the size and (b) for the profile of the shape. The first measurement uses a system of peak detecting registers, described as excursion registers (35) to record the maximum horizontal (right-most) and vertical (topmost and bottom-most) extents of the shape, note that the left-most extent corresponds to the X value

(of the Y axis) of the found coordinate. Thus the final values within the excursion registers (35) represent the size of the enclosing rectangle for the character shape. The second measurement uses a pair of random access memories (36) and (37) to record the left most and right-most horizontal pixel coordinates for every line (one pixel width) of the shape addressed by the vertical coordinate. The left and right pixels are indicated by 'L' and 'R' respectively in Figure 8C and represent the left and right profiles of the character shape. The character may now be extracted from the bit map memory by performing a raster scan (of the bit map memory) for the enclosing rectangle in the direction left to right, moving downward, allowing only those pixels whose coordinates are within the range enclosed by the left and right profiles of the character. This is achieved by passing the coordinate values of the right and left profiles and the coordinates of the enclosing rectangle to an extract control circuit (38). This scan will have the effect of removing any intrusions (within the enclosing rectangle) due to overlapping characters. The resultant extracted shape will have the form shown in Figure 8D which shows the required result of removing the intruding portions of the neighbouring letter "o".

The "aligned coordinate" is determined as the topmost and left-most coordinate of the enclosing rectangle for that character as illustrated in Figure 9. The "aligned coordinate" is loaded into the aligned coordinate register (39).

The extracted shape is then fed into the normalise and randomise system function (11) at the same time a message is sent to the computer control system (13) (step 109, Fig.3) this message contains the size limits for the extracted shape and the "aligned coordinate"

The aligned coordinate provides a datum for the character position and is used to reassemble (re-compose) the text and the page (of recognised characters).

The computer control system (13) assesses the enclosing rectangle for any of the following conditions:

(a) Too small.

(b) Too large.

(c) Aspect ratio (height to width) incorrect for a single character.

If (a) or (b) condition applies, the classification operation is aborted (step 112, Fig.3) and the pixel group comprising the extracted character is classed as unidentifiable, i.e. an unrecognised character. If (c) condition applies, the unrecognised pixel group block is divided into a series of sub-blocks, by estimation of the character boundaries within the pixel group block, each sub-block is submitted separately to the classifier (step 113 Fig.3).

The extracted (segmented) character shape is required to be "normalised" to a standard "enclosing rectangle" size, e.g. 32 x 32 pixels, prior to classification. Since the extracted character shape may be any size (in pixels) the normalisation can be achieved by initially scaling downwards in area by say 4:1, 16:1, 64:1 ratios, so that the scaled shape size is less than the required normalisation size and then using a look-up table approach to achieve the required normalisation result. Figure 10A illustrates the technique. The initial scaling (downwards) is achieved by the fixed scaling system (40) and the size "normalisation" by the variable scaling system (41).

An example of a variable scaling system (41) is shown in more detail in Figure 11. The system has

horizontal and vertical counters (42) and (43), driven by a clock which has a frequency chosen to suit the cycle time of the memories. The horizontal and vertical counters (42),(43) are a pair of counters, which count from zero to full house one's during the scaling operation. Horizontal and vertical size registers (44) and (45) are provided, each comprising a 5 bit register whose contents do not change during the scaling operation, having been (previously) set to the actual size of the character shape within the bit map memory. The values held in the size registers are actually one less than the size of the shape to be scaled, i.e. a size register value of 00011 binary (3 decimal) indicates, to the scaling system, that the shape has a size of four pixels in that particular direction, horizontal or vertical. The horizontal and vertical scaling memories (46),(47) connected with the respective counters and registers (42)-(45) are identical and conveniently consist of 1024 by 5 bit static RAM's (Random Access Memories), although a read only form of memory could also be used. Scaling tables would be written to RAM's at power-up, whereas read only memories would have scaling tables already "burnt in". Each 1024 by 5 bit scaling memory has a ten bit address, made up from the five bits (each) of the appropriate counter and size registers, (42),(44) and (43),(45). The 5 bits from the counter will count up from zero as the scaling is performed while the 5 bits from the size register remain constant so that, from the scaling tables in Figure 12, a sequence of pixel numbers may be generated. These are used as pixel pick-up addresses by the bit map memory that holds the shape being scaled. For any particular address presented to the scaling memory, a 5 bit data word will be available at the data out terminals. These are

referred to as the modified x and y addresses, x and y corresponding to horizontal and vertical pixel coordinates within the shape in the bit map memory. The two groups of modified addresses are used as a ten bit address for the bit map memory and the overall effect has been to generate an address sequence which has been pixel by pixel adjusted by the scaling memories in such a way that pixels are picked up from the bit map in a repeated fashion. The extent to which the pixels are repeated is a function of the size of the shape as defined by the values in the size registers. If the size registers are set to 11111, then the shape within the bit map is already at maximum and the particular sequence of modified addresses,(produced by the scaling tables) will be a count that is identical to the output from the counter for that axis.

Referring to the scaling table (Fig.12) entries, e.g. the horizontal list of numbers, for SIZE = 31 it will be seen that there are no duplicated pixel address values. The sequence for SIZE = 31 is itself a binary count from 0 to 31. In this instance the scaling tables will have had no effect but for all other sizes less than 11111 (31 decimal) the shape will be scaled up to a shape which is 32 pixels wide by 32 pixels high. The scaling tables Fig.12 are to be further described.

The output memory (48) is a static Random Access Memory providing storage of 1024 by 1 pixels. During the scaling operation the ten bit address to this output memory (48) is driven by the two counters (42), (43), horizontal and vertical, from zero to full house ones. Every pixel is thus addressed once, with the black or white value written into the particular location, being derived from the pixels stored in the bit map, that had been addressed by the address that has been modified as described.

Referring again to Figure 10A, the variable scaling algorithm in relation to the operation of the scaling tables, may be represented as follows: Variable scaling, on a given axis, is determined by the maximum size of the intermediate block (Fig 10A), along that axis. If N is the maximum (pixel) excursion in the intermediate block, then the 'Y' axis of the table (labelled Size S) equals (N-1). The 'X' axis of the table (labelled P) is the Pixel Number (P) for the final pixel block, i.e. P goes from 0 to 31, for a 32x32 normalised pixel block. The table value, as selected by the 'X' and 'Y' table coordinates, is the Pixel Number M in the intermediate block. Referring to Figure 10B, in conjunction with the scaling tables Fig.12, if the maximum (pixel) excursion along an axis in the intermediate block is 25, then the Size S equals (N-1) is 24 and the pixel states (black or white), in the final pixel block P, are determined from the pixel numbers (locations) derived from the tables. For the example shown in Fig.10B:- the pixel state (black or white) at the final pixel location P=10 will be the pixel state at the intermediate pixel location M=7. Similarly for P=24 the pixel state will be that at the intermediate location M=19.

The scaled "normalised" character shape is now presented to the randomisation function (Fig.10A). The randomisation function generates pseudo-random n-tuples by use of another look-up table. The requirement is that the normalised (32x32) pixel group block must be mapped into a series of n-tuples such that:

(a) The grouping of pixels (n-tuples) is selected on a random basis.

(b) The selection of the pixels is such that no pixel appears in more than one n tuple and then only once.

(c) The pixel block (32x32) must be completely mapped, i.e.every pixel must appear in the set of n-tuples.

This requirement for random n-tuples is dealt with in the referenced papers on N-tuple technology. Referring to Figure 13A, this represents the 32x32 pixel block mapped into 128 separate 8-tuples. The initial relationship between the pixels selected to form 8-tuples is random but once this random selection has been chosen, it remains unchanged. A look-up table can be constructed to map, a given pixel location, to a given bit number, in a given 8-tuple. For the example shown in Fig 13A, if the 32x32 pixel block coordinates are set such that coordinate 0,0 corresponds to the top left hand corner of the map then the mapping illustrated would correspond to the (part) table of Figure 13B. Such a table could be constructed which would identify each bit in each 8-tuple with a specific pixel location in the pixel block. The bit value would be '1' or '0' to correspond with the black or white state of the pixel so located.

The preceding descriptions (related to Figure 10A) are not intended to imply that the described functions, of Fixed Scaling, Variable Scaling (Normalisation), Randomising, are separate serial activities. They have been so described for ease of understanding. The normalise and randomise function, (11) is such that the three functions (described) are carried out in an overlapping sequential manner so as to appear as a single integrated function.

The look-up tables may conveniently reside in:-

either (a) non volatile memory (e.g. PROM, PAL etc).

or (b) volatile memory (e.g. RAM), initialised by software on power-up of the machine.

That is the same approach to look-up tables can be made

as previously described for combinational transition functions.

An additional advantage of this technique is that the various areas may ultimately be implemented as PAL based functions, which offer protection of the design against unauthorised copying, since such (PAL based) functions are very much more difficult to reverse engineer than PROM based functions.

The final operation is to load the N-tuple buffer input to the classification system (12) and to send a "finished" message to the computer control system (13) (step 114, Fig.3).

The computer control system (13) now decides as to whether to proceed with the classification process, or to abort for the reasons previously stated, or to go into a classification sub-routine.

The classification (normal routine) is initiated at step 115, Fig.3 by the computer control system (13). The classification system (12) as previously mentioned is a synchronous state machine. The approach is similar to that already described in relation to the synchronous state machine for the segmentation system. That is combinational transition functions are used to define the conditions and sequence of the state machine.

The methods of operation of random n-tuple classifiers are described in the reference papers on N-tuple technology. The "classifier" is pre-trained with the range of patterns or classes it is required to recognise. When an unknown pattern is entered, the classifier responds with a ranking list of the classes, i.e. of the "most like" scores relative to the training set. The N-tuple method (technique) is essentially a means of comparing the unknown pattern with the range

of patterns already "learnt" by the classifier, so that the classifier can make "most like" decisions. The top ranked (score) would (normally) be selected as that representing the pattern. In the preferred embodiment, this selection would also be dependent on:

(a) The score relative to some threshold A above which the character is identified (classified).

(b) The score relative to some threshold B below which the character is not identified.

(c) The Ranking Order of the scores, i.e. the relative discrimination between the top ranked class and the next highest scoring class or classes.

The classification system (12) is shown in more detail in Figure 14. The mode of operation of the classification system is illustrated in Figure 15.

The classification system comprises an n-tuple counter (50) and a (Class) group counter (51). These counters are driven by the same system clock which drives the scaling system previously described. The n-tuple and group counters (50),(51) comprise a seven bit counter and a three bit counter, respectively connected as a ten bit counter. This counter counts from zero to full house one's during the response calculation operation. Initially the counters are set to zero (step 200, Figure 15). The output from the n-tuple counter (50) is a number which is used as an address for an n-tuple memory (49). The seven bits are used to sequentially address the 128 n-tuples stored within that memory.

The n-tuple memory (49) will have previously been loaded from the normalise and randomise system function (11) and will contain a random n-tuple pattern of bits that represent the normalised shape, that has been extracted from the bitmap memory (7). The n-tuple memory (49) consists of a static Random Access Memory

with a storage capacity of 128 eight bit values, these values being the n-tuples that make up the shape to be recognised (i.e. n=8).

The n-tuples are addressed sequentially by the incrementing n-tuple counter (50) and the eight bit values of those n-tuples are presented to a discriminator memory (53), as addresses which are combined with both the seven bit output of the n-tuple counter (50) and the four bit output from the group counter (51) to produce a 19 bit address that is used by the discriminator memory (53).

Note: It is assumed that the discriminator memory has previously been loaded with the responses generated from training data as previously described and as referenced in the papers that describe the operation of an N-tuple based recognition system.

The discriminator memory (53) is a Random Access Memory constructed from Dynamic Random Access Memory elements which are organised, for the purposes of a parallel response discriminator, as an eight bit wide data bus memory system.

During the calculation of the responses, the values read from the discriminator memory (53) are interpreted as single bit responses (step 202, Fig.15), these are required to be summed in order to produce total responses for all classes that are being tested for possible recognition. In order to provide these summed totals, a collection of eight bit counters or incrementors (54) are connected to the data output terminals of the discriminator in such a fashion that they will increment, or count up by one, if the particular discriminator data bit corresponding to that particular value of the n-tuple is a logic one. If the discriminator provides a logic zero then the up counter

will ignore it and retain its current value. All of the incrementors (54) are cleared to zero (step 201, Fig.15) at the beginning of every group, i.e. when the value in the n-tuple counter goes from 1111111 binary (31 decimal) to zero and the group counter increments by one. This initialises the incrementors (54) ready to produce response sum totals for the eight sub classes that constitute the next group.

Before the n-tuple counter begins its incrementing sequence from zero to 31 decimal, a class counter (55) is used to read the eight bit values that have accrued in the response incrementors (54) (step 204, Fig.15) and to write them into a table of responses stored in a responses memory (56) (step 205, Fig.15). The responses memory (56) consists of a static Random Access Memory, organised according to the number of classes of recognition (classification).

At the completion of the classification function (step 206, Fig.15) a message is sent to the computer control system (13), providing classification data(step 116, Fig.3). The computer control system (13) then proceeds with the initial post-processing (stage 1) (to be described) and recommences the segmentation routine (step 117, Fig.3), i.e. returns to step 108, Fig.3.

The segmentation/classification sequence continues until all the characters are classified, i.e. until all the patterns within the image bitmap (7) have been extracted, segmented, normalised and classified. When the classification "finish" is reached (step 118, Fig.3), the computer control system (13) continues the post-processing (stage 2).

The initial post-processing (stage 1) is to check for items, such as punctuation, known ambiguities, nonsense (based on known response values), invalid

classes etc. and to complete the identification of the character.

The final post-processing (stage 2) is to reassemble the character data into a "format" with the classification "errors" as (I) below highlighted. Classification errors are:

(I) Reject error, where the classifier is unable to make a true decision.

(II) Substitution error, where the classifier makes a wrong decision.

In the case of (I) above, it is possible, by known computing means, to arrange to output from the recognition unit, for subsequent display, the entire pixel group representing the character, this allows for a human interrogation (intervention). To allow for this facility the output from the stage 1 post-processing should be loaded into a "shape buffer" memory store.

In order to ensure a correct ordering of each pattern, as it is classified the stage 1 post-processing software has to arrange to "tag" each result with the image map location data previously received (steps 107,109,Fig.3); this information may then be used, in the embodiment for text recognition, to recompose the page and ensure the correct ordering of the recognised characters, as described for stage 2 post-processing.

In the event that a result from the stage 1 post-processing is that no one character class is sufficiently clear, the computer control system(13) may decide to require that the character is reclassified, e.g. by presentation to a sub-set of classes as previously explained. It should also be noted that the order in which the discriminator memory (53) is accessed may be arranged in a special form, again as

previously described. For example, in the case of English language text, the first classes accessed may comprise the vowels. In this case, the computer control system (13) may compare each response against predetermined recognition criteria and as soon as those criteria are satisfied will terminate further classification.

Post-processing can be used to carry out other functions, to improve the error rate and to provide special facilities, for example:

(a) Minimise errors due to case confusion.

(b) Minimise errors due to alpha/numeric confusion.

(c) Allow the definition of selected fields within an image and select those fields only to be processed.

(d) Allow selected fields to be defined as alpha or numeric or mixed.

(e) Apply additional rules to pixel groups of patterns which are not recognised, or poorly discriminated.

(f) Apply dictionary and/or context correction techniques to reduce errors.

(g) Ensure the classified patterns are ordered to a predetermined format, as appropriate to the application.

CLAIMS

1. Image recognition apparatus comprising a first synchronous state machine for segmenting a number of images defined in bit map form into separate pixel groups; and a second synchronous state machine to which each pixel group is applied for classification.

2. Apparatus for recognizing images represented by respective digital pixel groups, the apparatus comprising an N-tuple classifier including a number of discriminators each adapted to recognise a respective class of a predetermined group of classes and to which the pixel groups are presented, the apparatus being arranged to present each pixel group to the discriminators in a predetermined sequence; and recognition means for monitoring the output of the discriminators and for terminating the presentation of the pixel group to the classifier as soon as the output from a discriminator satisfies a recognition condition.

3. Apparatus according to claim 1 and claim 2.

4. A method of recognizing images represented by respective digital pixel groups, the method comprising presenting each pixel group to an N-tuple classifier having a number of discriminators each adapted to recognise a respective class of a predetermined group of classes, is characterised in that each pixel group is presented to the discriminators in a predetermined sequence; and in that as soon as the output from a discriminator satisfies a recognition condition, the presentation of the pixel group to the classifier is terminated.

5. A method according to claim 4, comprising comparing the output from each discriminator with a threshold, the recognition condition being satisfied when the threshold is exceeded.

6. A method according to claim 4, wherein each pixel group is presented to the discriminators in the order of frequency of occurrence of the classes represented by the discriminators.

7. A method according to claim 4, wherein the discriminator or discriminators to which each pixel group is applied are chosen in accordance with the location of the pixel group defining the images within the context of the previously detected images.

8. A method for recognising images represented by respective digital pixel groups, the method comprising presenting each pixel group to an N-tuple classifer having a number of discriminators each adapted to recognise a respective class of a predetermined group of classes characterised in that if none of the discriminator outputs satisfies a recognition condition but it is determined that the pixel group defines an image falling within a group of the classes, the method further comprises presenting a portion of the pixel group to a subsidiary N-tuple classifier having a number of subsidiary discriminators each adapted to recognise a respective portion of the group of classes.

9. A method according to claim 8, further comprising storing data defining the recognized class of the image represented by the pixel group.

10. A method according to claim 8 or claim 9, wherein each pixel group is presented simultaneously to groups of two or more discriminators in the classifier and, where appropriate, the subsidiary classifier.

11. Apparatus for recognizing images represented by respective digital pixel groups, the apparatus comprising an N-tuple classifier having a number of discriminators each adapted to recognise a respective class of a predetermined group of classes and to which each pixel group is presented; recognition means for monitoring the

outputs of the discriminators; and a subsidiary N-tuple classifier having a number of subsidiary discriminators each adapted to recognise a respective class of a predetermined group of classes defining portions of a respective group of images, the recognition means being adapted to present a portion of a pixel group to the subsidiary classifier if it is determined that the discriminator outputs do not satisfy a recognition condition but the discriminator outputs define an image falling within the group of classes.

12. Apparatus according to claim 11, further comprising storage means for storing data defining the recognized class of the image represented by the pixel group.

13. A method of segmenting images represented in bit map form, the method comprising scanning the bit map to determine the maximum extents of an image in first and second orthogonal directions and recording for each scan line in the first direction the coordinates of the extreme pixels of the image in the second orthogonal direction; and selecting as defining an image only those pixels within a rectangle defined by the previously determined extents and falling within the previously determined extreme pixel coordinates.

14. A method according to claim 13, wherein the scanning of the bit map is carried out in a series of horizontally spaced, vertical scan lines and this leads to the ability to compensate for skew from a knowledge of the line spacing or pitch deduced from a histogram analysis of the page of text.

15. A method according to claim 13 or claim 14, wherein the selecting step comprises scanning the bit map in a series of lines extending in a second orthogonal direction and spaced apart in the first orthogonal direction, each line having a length corresponding to the

distance between the respective extreme pixel coordinates.

16. Apparatus for segmenting images represented in bit map form, the apparatus comprising scanning the bit map to determine the maximum extents of an image in first and second orthogonal directions and recording for each scan line in the first direction the coordinates of the extreme pixels of the image in the second orthogonal direction; and selecting as defining an image only those pixels within a rectangle defined by the previously determined extents and falling within the previously determined extreme pixel coordinates.

17. A method of segmented images represented in bit map form, the method comprising

a) scanning the bit map to detect a shape which may comprise an image;

b) recording the location of those pixels in the bit map which define the detected shape;

and repeating the steps a) and b) to locate other images while ignoring in step a) each pixel whose location has been recorded in a step b).

18. A method according to claim 16, wherein step b) comprises providing a second bit map coterminous with the bit map defining the images, and recording in the second bit map those pixels which have been found during the scanning step to correspond to a detected shape.

19. Apparatus for segmenting images represented in bit map form, the apparatus comprising scanning means for scanning the bit map to detect a shape which may comprise an image; and a memory for recording the location of those pixels in the bit map which define the detected shape, whereby the scanning means only responds to those pixels of the bit map whose locations have not been recorded in the memory.

Fig.1.

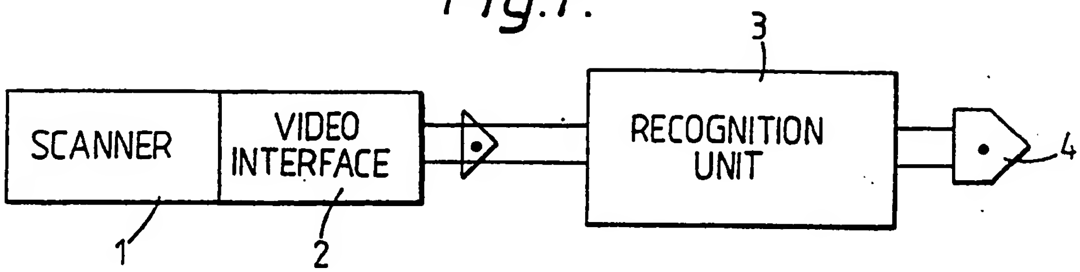
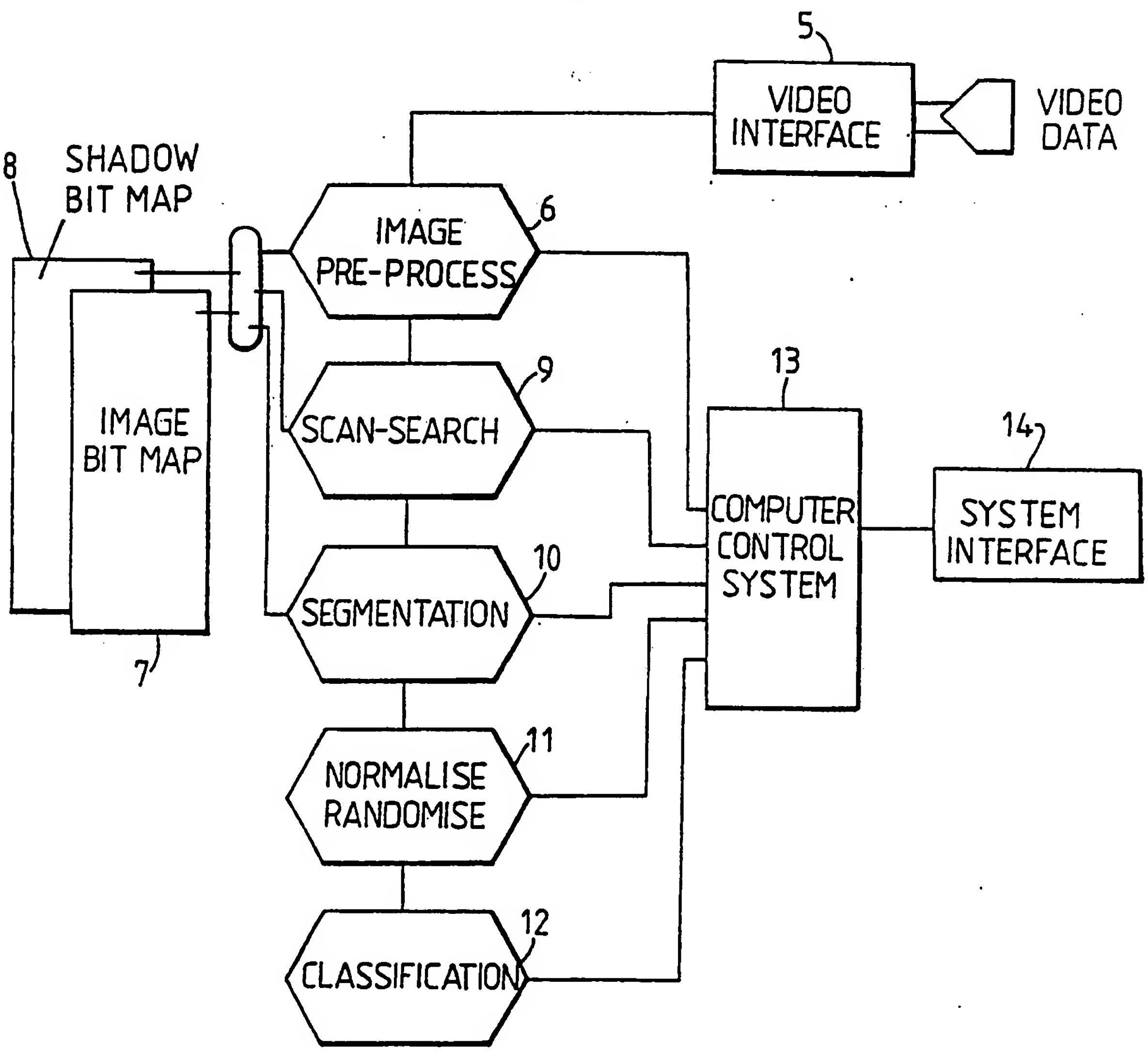

3
SCANNER
VIDEO INTERFACE
RECOGNITION UNIT
4
1
2

Fig.2.


5
VIDEO INTERFACE
VIDEO DATA
8
SHADOW BIT MAP
IMAGE PRE-PROCESS
6
IMAGE BIT MAP
SCAN-SEARCH
9
13
COMPUTER CONTROL SYSTEM
14
SYSTEM INTERFACE
SEGMENTATION
10
7
NORMALISE RANDOMISE
11
CLASSIFICATION
12

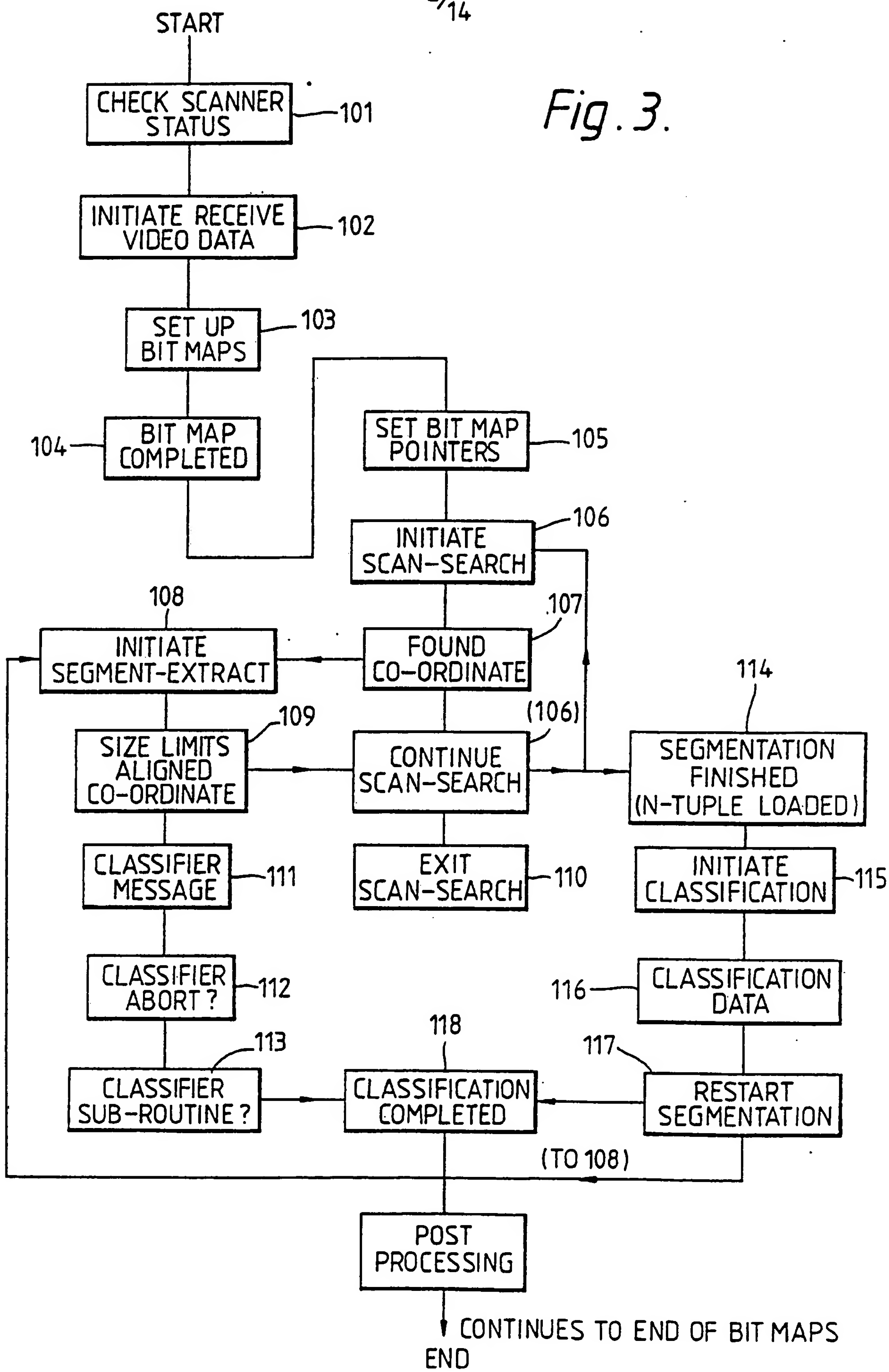

START
CHECK SCANNER STATUS
101
INITIATE RECEIVE VIDEO DATA
102
SET UP BIT MAPS
103
BIT MAP COMPLETED
104
SET BIT MAP POINTERS
105
INITIATE SCAN-SEARCH
106
INITIATE SEGMENT-EXTRACT
108
FOUND CO-ORDINATE
107
SIZE LIMITS ALIGNED CO-ORDINATE
109
CONTINUE SCAN-SEARCH
(106)
SEGMENTATION FINISHED (N-TUPLE LOADED)
114
CLASSIFIER MESSAGE
111
EXIT SCAN-SEARCH
110
INITIATE CLASSIFICATION
115
CLASSIFIER ABORT ?
112
CLASSIFICATION DATA
116
CLASSIFIER SUB-ROUTINE ?
113
CLASSIFICATION COMPLETED
118
RESTART SEGMENTATION
117
(TO 108)
POST PROCESSING
CONTINUES TO END OF BIT MAPS
END

Fig. 3.

Fig.4.

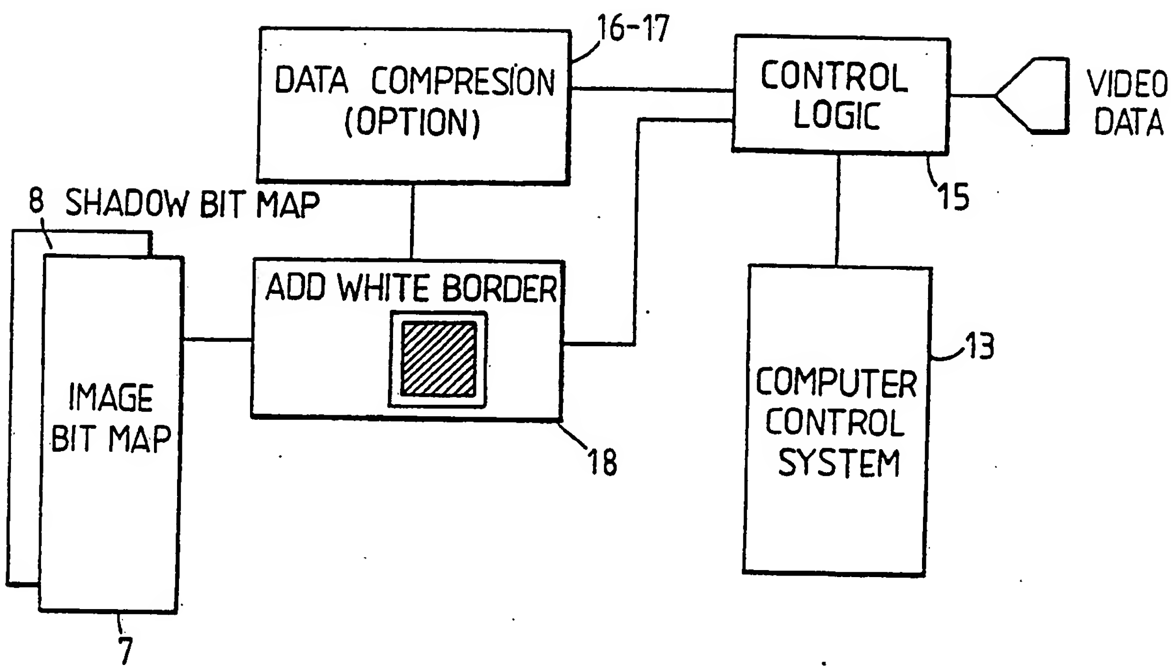

DATA COMPRESION (OPTION)
16-17
CONTROL LOGIC
VIDEO DATA
8 SHADOW BIT MAP
15
ADD WHITE BORDER
IMAGE BIT MAP
13
COMPUTER CONTROL SYSTEM
18
7

Fig.5.

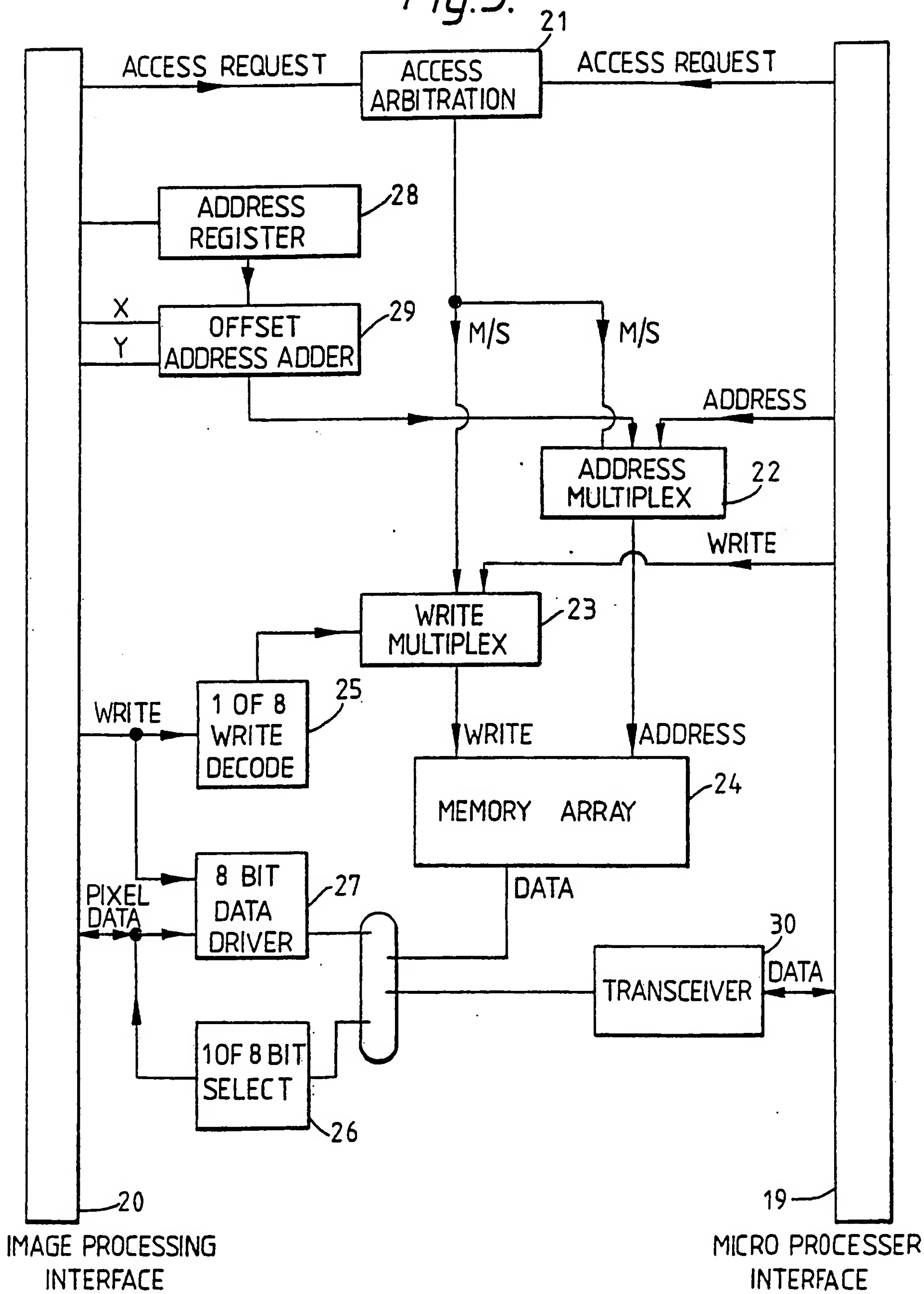

21
ACCESS REQUEST
ACCESS ARBITRATION
ACCESS REQUEST
ADDRESS REGISTER
28
X
Y
OFFSET ADDRESS ADDER
29
M/S
M/S
ADDRESS
ADDRESS MULTIPLEX
22
WRITE
WRITE MULTIPLEX
23
WRITE
1 OF 8 WRITE DECODE
25
WRITE
ADDRESS
MEMORY ARRAY
24
PIXEL DATA
8 BIT DATA DRIVER
27
DATA
30
DATA
TRANSCEIVER
1OF 8 BIT SELECT
26
20
19
IMAGE PROCESSING INTERFACE
MICRO PROCESSER INTERFACE

Fig.6.

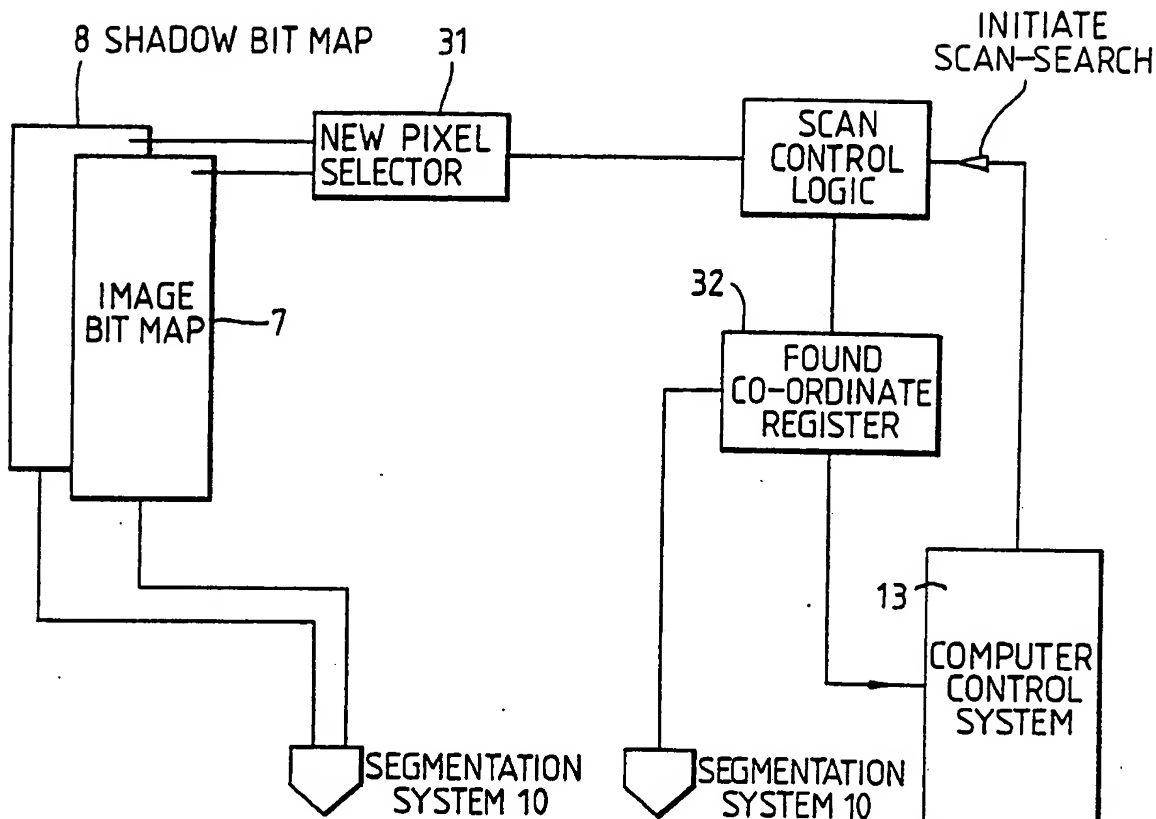

8 SHADOW BIT MAP
31
NEW PIXEL SELECTOR
INITIATE SCAN-SEARCH
SCAN CONTROL LOGIC
IMAGE BIT MAP
7
32
FOUND CO-ORDINATE REGISTER
13
COMPUTER CONTROL SYSTEM
SEGMENTATION SYSTEM 10
SEGMENTATION SYSTEM 10

Fig.7.

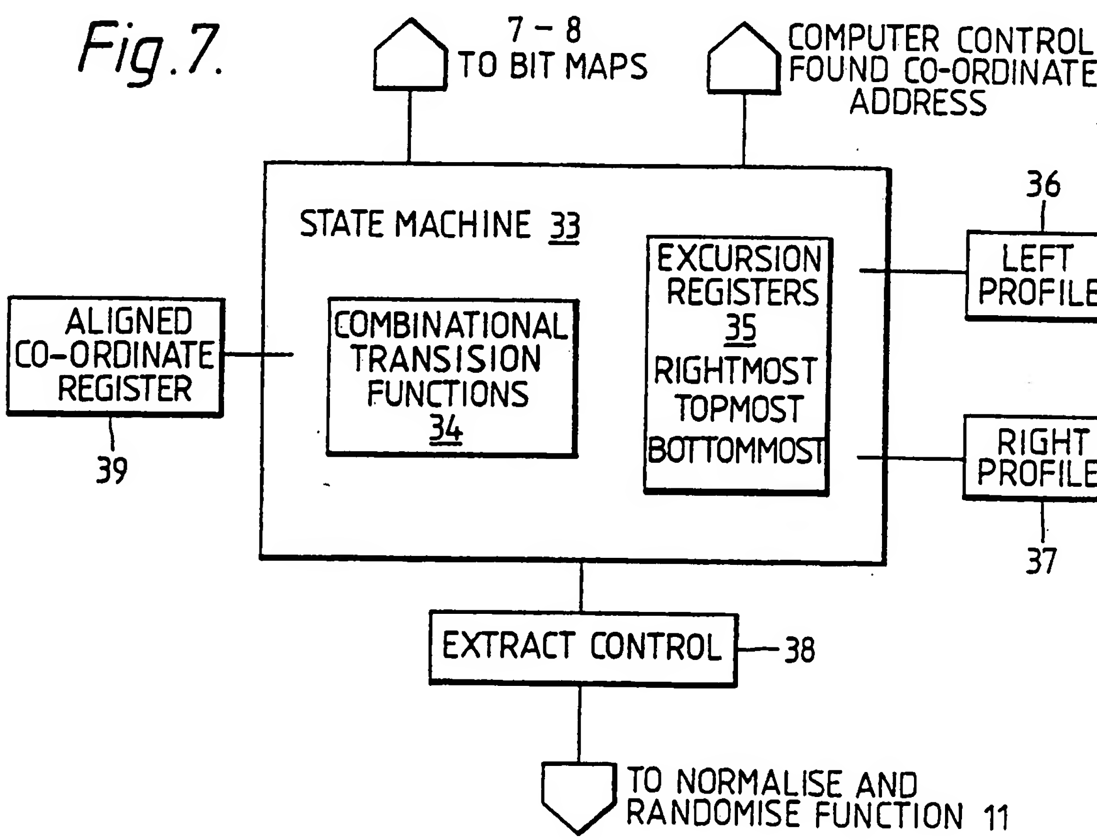
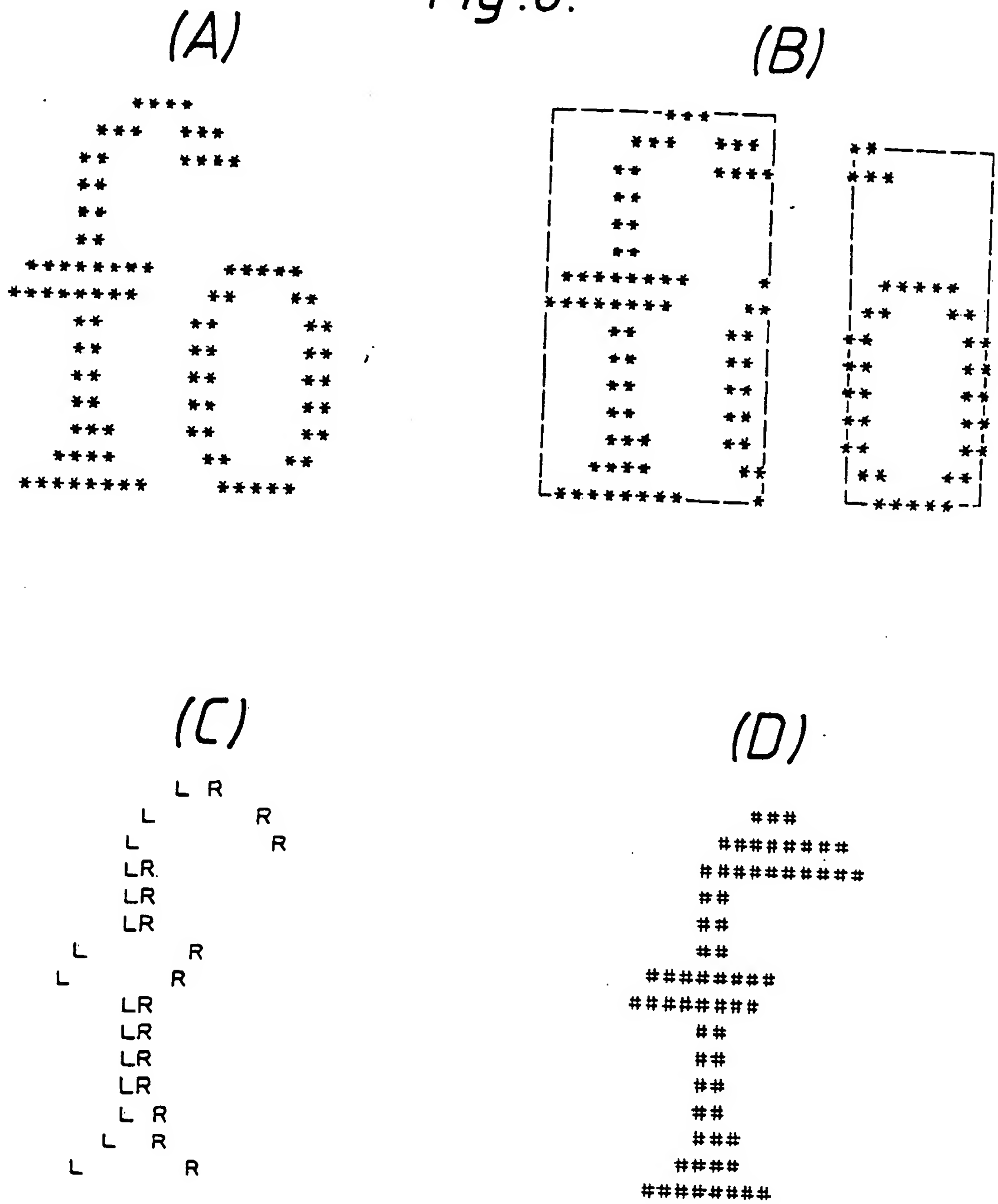

7 - 8 TO BIT MAPS
COMPUTER CONTROL FOUND CO-ORDINATE ADDRESS
STATE MACHINE 33
36
EXCURSION REGISTERS 35 RIGHTMOST TOPMOST BOTTOMMOST
LEFT PROFILE
ALIGNED CO-ORDINATE REGISTER
COMBINATIONAL TRANSISION FUNCTIONS 34
39
RIGHT PROFILE
37
EXTRACT CONTROL
38
TO NORMALISE AND RANDOMISE FUNCTION 11

Fig. 8.



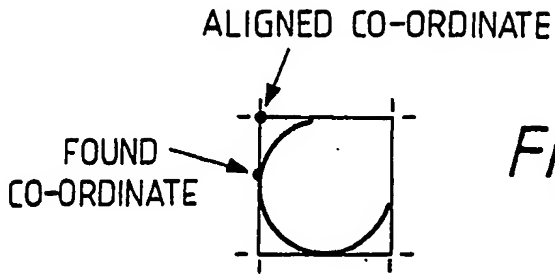
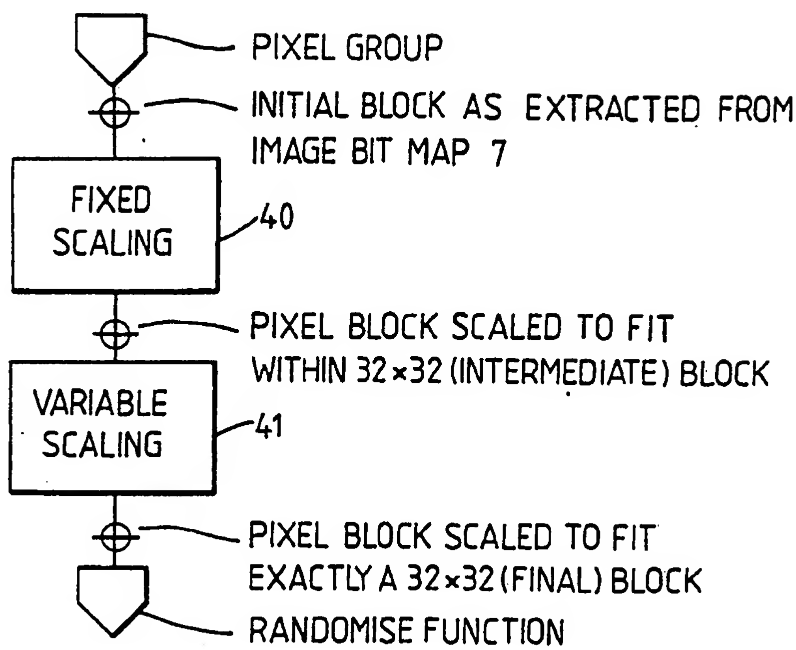
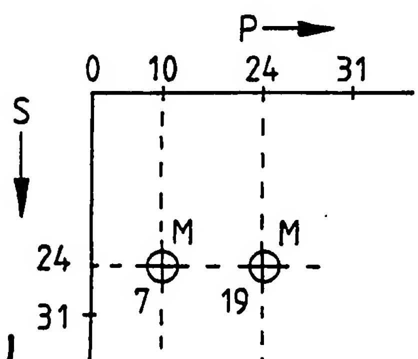


Fig. 9.

Fig. 10.



(A)



(B)

Fig.11.

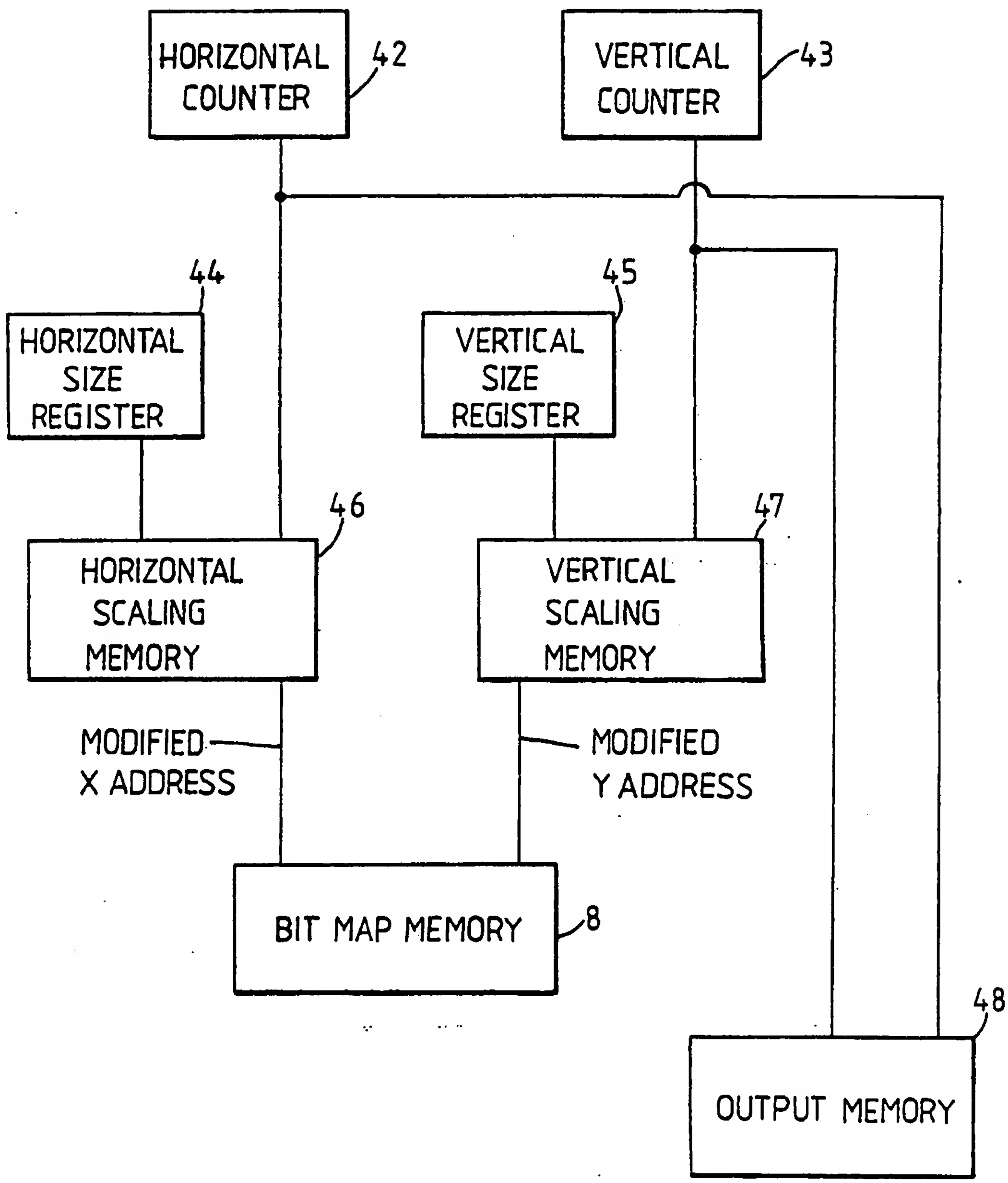

HORIZONTAL COUNTER
42
VERTICAL COUNTER
43
44
HORIZONTAL SIZE REGISTER
45
VERTICAL SIZE REGISTER
46
HORIZONTAL SCALING MEMORY
47
VERTICAL SCALING MEMORY
MODIFIED X ADDRESS
MODIFIED Y ADDRESS
BIT MAP MEMORY
8
48
OUTPUT MEMORY

Fig. 12.

SCALING TABLE EXAMPLE FOR A 32 × 32 FINAL SHAPE

PIXEL NUMBER P →

SIZE S ↓	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
4	0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2
5	0	0	0	0	0	0	1	1	1	1	1	2	2	2	2	2
6	0	0	0	0	0	1	1	1	1	1	2	2	2	2	3	3
7	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
8	0	0	0	0	1	1	1	1	2	2	2	3	3	3	3	4
9	0	0	0	0	1	1	1	2	2	2	3	3	3	4	4	4
10	0	0	0	1	1	1	2	2	2	3	3	3	4	4	4	5
11	0	0	0	1	1	1	2	2	3	3	3	4	4	4	5	5
12	0	0	0	1	1	2	2	2	3	3	4	4	4	5	5	6
13	0	0	0	1	1	2	2	3	3	3	4	4	5	5	6	6
14	0	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7
15	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7
16	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7
17	0	0	1	1	2	2	3	3	4	5	5	6	6	7	7	8
18	0	0	1	1	2	2	3	4	4	5	5	6	7	7	8	8
19	0	0	1	1	2	3	3	4	5	5	6	6	7	8	8	9
20	0	0	1	1	2	3	3	4	5	5	6	7	7	8	9	9
21	0	0	1	2	2	3	4	4	5	6	6	7	8	8	9	10
22	0	0	1	2	2	3	4	5	5	6	7	7	8	9	10	10
23	0	0	1	2	3	3	4	5	6	6	7	8	9	9	10	11
24	0	0	1	2	3	3	4	5	6	7	7	8	9	10	10	11
25	0	0	1	2	3	4	4	5	6	7	8	8	9	10	11	12
26	0	0	1	2	3	4	5	5	6	7	8	9	10	10	11	12
27	0	0	1	2	3	4	5	6	7	7	8	9	10	11	12	13
28	0	0	1	2	3	4	5	6	7	8	9	9	10	11	12	13
29	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
30	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
31	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

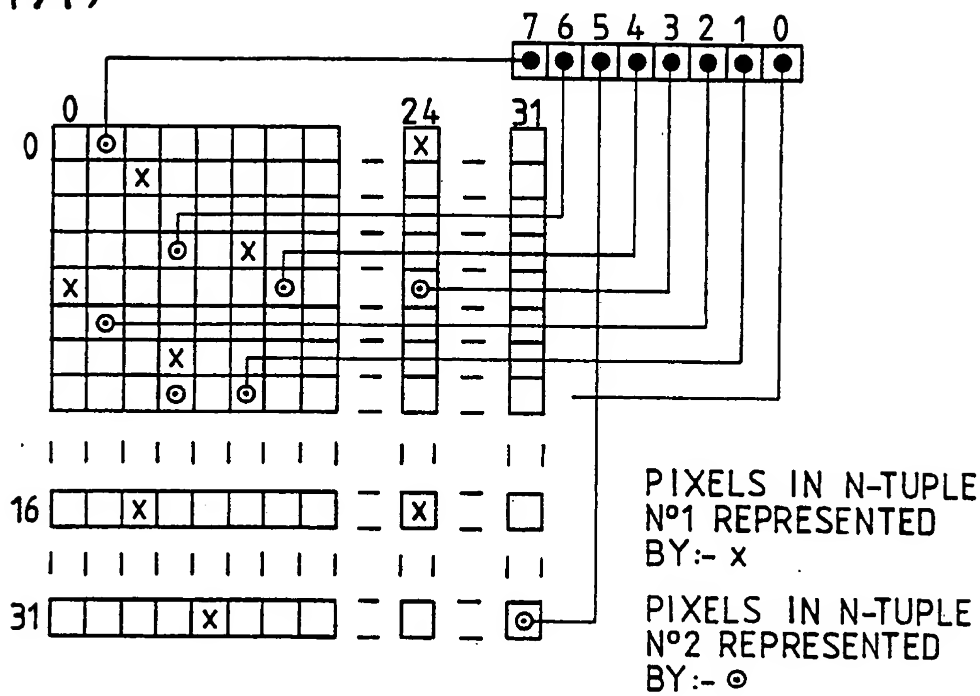
Fig.12(cont)

P →

SIZE : (S ↓)	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
4	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4
5	3	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5
6	3	3	3	4	4	4	4	5	5	5	5	5	6	6	6	6
7	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7
8	4	4	5	5	5	5	6	6	6	7	7	7	7	8	8	8
9	5	5	5	5	6	6	6	7	7	7	8	8	8	9	9	9
10	5	5	6	6	6	7	7	7	8	8	8	9	9	9	10	10
11	6	6	6	7	7	7	8	8	9	9	9	10	10	10	11	11
12	6	6	7	7	8	8	8	9	9	10	10	10	11	11	12	12
13	7	7	7	8	8	9	9	10	10	10	11	11	12	12	13	13
14	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14
15	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15
16	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16
17	9	9	10	10	11	11	12	12	13	14	14	15	15	16	16	17
18	9	10	10	11	11	12	13	13	14	14	15	16	16	17	17	18
19	10	10	11	11	12	13	13	14	15	15	16	16	17	18	18	19
20	10	11	11	12	13	13	14	15	15	16	17	17	18	19	19	20
21	11	11	12	13	13	14	15	15	16	17	17	18	19	19	20	21
22	11	12	12	13	14	15	15	16	17	17	18	19	20	20	21	22
23	12	12	13	14	15	15	16	17	18	18	19	20	21	21	22	23
24	12	13	14	14	15	16	17	17	18	19	20	21	21	22	23	24
25	13	13	14	15	16	17	17	18	19	20	21	21	22	23	24	25
26	13	14	15	16	16	17	18	19	20	21	21	22	23	24	25	26
27	14	14	15	16	17	18	19	20	21	21	22	23	24	25	26	27
28	14	15	16	17	18	19	19	20	21	22	23	24	25	26	27	28
29	15	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
30	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Fig.13

(A)



(B)

EXAMPLE FOR N-TUPLE N°2 :-

N-TUPLE N°	BIT N°	CO-ORDINATE
2	0	3,7
"	1	5,7
"	2	1,5
"	3	24,4
"	4	6,4
"	5	31,31
"	6	3,3
"	7	1,0

Fig. 14.

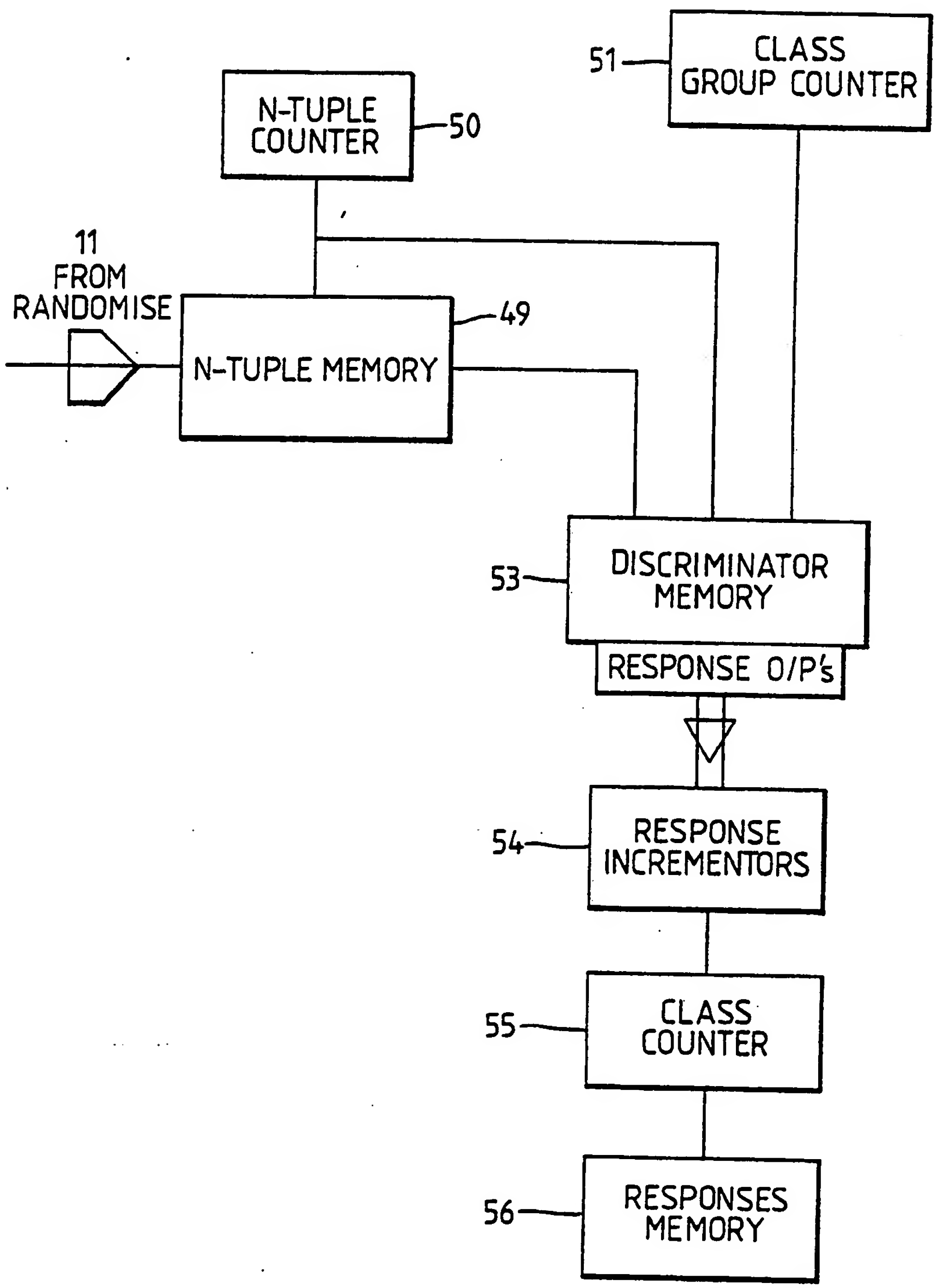

CLASS GROUP COUNTER
51
N-TUPLE COUNTER
50
11
FROM RANDOMISE
N-TUPLE MEMORY
49
DISCRIMINATOR MEMORY
53
RESPONSE O/P's
RESPONSE INCREMENTORS
54
CLASS COUNTER
55
RESPONSES MEMORY
56

Fig. 15.

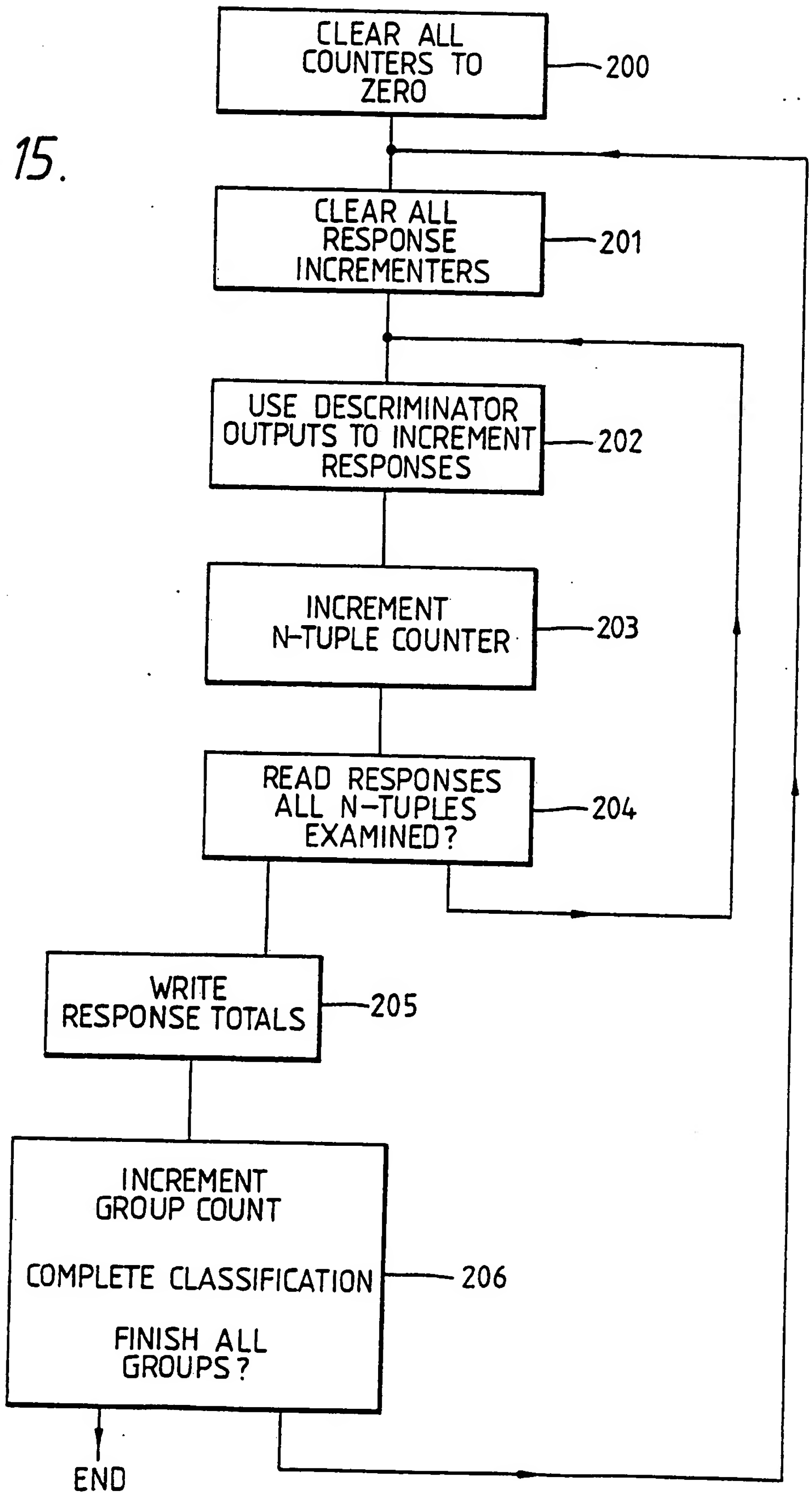
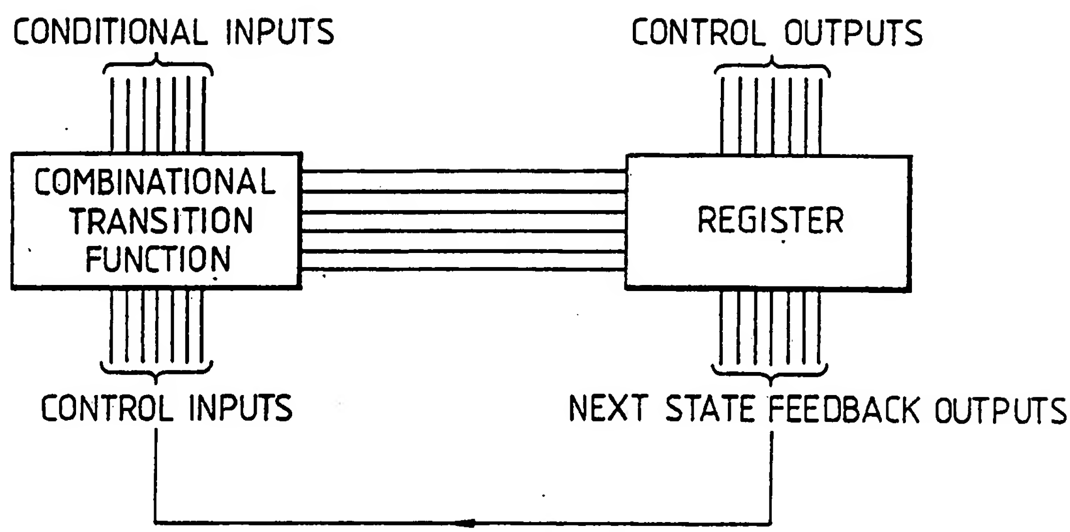

CLEAR ALL COUNTERS TO ZERO
200
CLEAR ALL RESPONSE INCREMENTERS
201
USE DESCRIMINATOR OUTPUTS TO INCREMENT RESPONSES
202
INCREMENT N-TUPLE COUNTER
203
READ RESPONSES ALL N-TUPLES EXAMINED?
204
WRITE RESPONSE TOTALS
205
INCREMENT GROUP COUNT
COMPLETE CLASSIFICATION
FINISH ALL GROUPS?
206
END

Fig.16.


CONDITIONAL INPUTS
CONTROL OUTPUTS
COMBINATIONAL TRANSITION FUNCTION
REGISTER
CONTROL INPUTS
NEXT STATE FEEDBACK OUTPUTS